

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 06-019791

(43)Date of publication of application : 28.01.1994

(51)Int.Cl. G06F 12/08
G06F 13/00
G06F 15/16

(21)Application number : 05-091945

(71)Applicant : SUN MICROSYST INC

(22)Date of filing : 29.03.1993

(72)Inventor : HAMILTON GRAHAM
NELSON MICHAEL N

(30)Priority

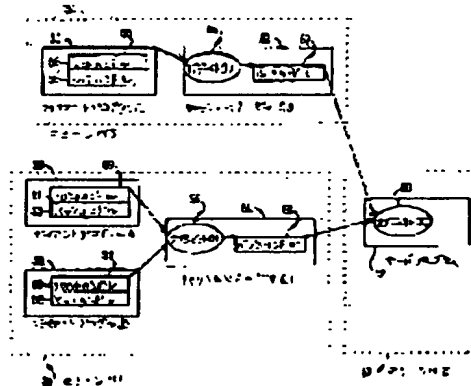
Priority number : 92 858788 Priority date : 27.03.1992 Priority country : US

(54) METHOD FOR USING LOCAL CACHE MANAGER AND DEVICE FOR THE SAME

(57)Abstract:

PURPOSE: To provide a means for effectively using a software caching service as much as possible in an object-oriented distributed computer system.

CONSTITUTION: The client program method and device suitable to an object-oriented distributed arithmetic system is provided with a capability of communication and a capability of using reference for an object, and this is a method for increasing the gain of software caching. Also, a mechanism for referring to both a server and a local cache manager by using a network object handle (increased object handle) is developed. When the increased object handle is transmitted, increased object handle is automatically communicated for using a local cache manager 62 under the target environment (that is, on the same machine in which a client program for reception is present).



LEGAL STATUS

[Date of request for examination] 22.02.2000

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 3621432

[Date of registration] 26.11.2004

[Number of appeal against examiner's decision]

Searching PAJ

2/2 ページ

of rejection]

[Date of requesting appeal against examiner's
decision of rejection]

[Date of extinction of right]

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平6-19791

(43)公開日 平成6年(1994)1月28日

(51)Int.Cl. ⁶	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 12/08	3 1 0 Z	7608-5B		
13/00	3 5 7 Z	7368-5B		
15/16	3 7 0 Z	8840-5L		

審査請求 未請求 請求項の数 2 (全 13 頁)

(21)出願番号 特願平5-91945

(22)出願日 平成5年(1993)3月29日

(31)優先権主張番号 858, 788

(32)優先日 1992年3月27日

(33)優先権主張国 米国 (U S)

(71)出願人 591064003

サン・マイクロシステムズ・インコーポレ
ーテッドSUN MICROSYSTEMS, IN
CORPORATEDアメリカ合衆国 94043 カリフォルニア
州・マウンテンビュー・ガルシア アヴェ
ニュー・2550

(72)発明者 グラハム・ハミルトン

アメリカ合衆国 94303 カリフォルニア
州・パロ アルト・デビッド コート・
3143

(74)代理人 弁理士 山川 政樹

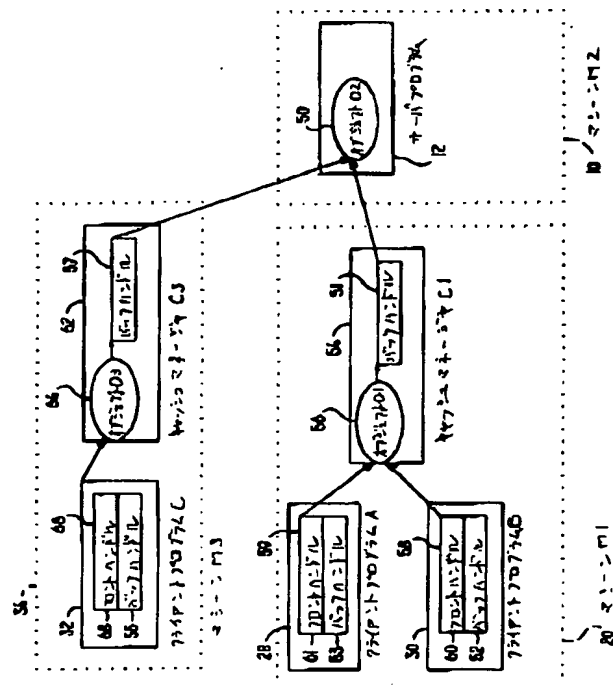
最終頁に続く

(54)【発明の名称】 ローカルキャッシュマネージャを用いる方法と装置

(57)【要約】

【目的】 本発明の目的は、オブジェクト指向型の分散型コンピュータシステムにおいて、ソフトウェアキャッシングサービスを可能な限り効果的に利用できる手段を提供することである。

【構成】 オブジェクト指向型の分散型演算システムに好適なクライアントプログラム方法および装置には、通信能力と、オブジェクトに対する参照を利用する能力とが設定されており、これは、ソフトウェアキャッシングの利益を増大する方法である。また、ネットワークオブジェクトハンドル（増大したオブジェクトハンドル）を用いて、サーバおよびローカルキャッシュマネージャの両方を参照できるメカニズムが開示されている。この増大されたオブジェクトハンドルが送信された場合に、そのターゲット環境の下で（即ち、受信用クライアントプログラムが存在している同一のマシン上で）、ローカルキャッシュマネージャを利用するために、この増大されたオブジェクトハンドルを自動的に交信する。



(2)

特開平6-19791

【特許請求の範囲】

【請求項1】 分散型演算環境の下でオブジェクト指向型プログラミングおよびソフトウェアキャッシング技術を用いるシステムで、受信用プログラムがターゲットオブジェクトを参照してローカルキャッシュマネージャを用いる方法において、

a) 前記ターゲットオブジェクトに対する参照と第1キャッシュマネージャに対する参照とを含む増大されたオブジェクトハンドルを受信する能力を前記受信用プログラムに創生する過程と、

b) 前記第1キャッシュマネージャに対する参照を、前記受信用プログラムに対する第2ローカルキャッシュマネージャに対する参照、または前記ターゲットオブジェクト自身に対する参照によって差し替えることによって、前記受信用プログラムに新規な増大オブジェクトハンドルを構成する能力を与える過程と、

c) 前記ターゲットオブジェクトのオペレーションを呼び出すために、前記受信用プログラムに、前記新規な増大オブジェクトハンドルを利用する能力を当てる過程とを有することを特徴とする方法。

【請求項2】 分散型演算環境の下でオブジェクト指向型プログラミングおよびソフトウェアキャッシング技術を使用するシステムで、ターゲットオブジェクトを参照して受信用プログラムがローカルキャッシュマネージャを使用する装置において、

a) 第1キャッシュマネージャに対する参照と、前記ターゲットオブジェクトに対する参照とに関連する増大されたオブジェクトハンドルを前記受信用プログラムが、他方のプログラムから受信する受信手段と、

b) 前記受信手段に結合され、前記第1キャッシュマネージャ用の参照を、前記受信用プログラムに対する第2のローカルキャッシュマネージャ用の参照に、または前記ターゲット自身への参照に変形するために前記増大したオブジェクトハンドルを変更する創作手段とを具え、前記プログラムが、前記新規な増大したオブジェクトハンドルを用いて、前記ターゲットオブジェクトのオペレーションを呼び出す能力を有することを特徴とする装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、分散型コンピュータシステム、クライアント・サーバ演算ならびにオブジェクト指向型プログラミングに関する。特に、クライアントプログラムを提供する方法および装置に関するもので、オブジェクトが利用されている各マシンにおいて、ソフトウェアキャッシングの完全な利益が得られるようなオブジェクトに関する参照を利用したり、これと通信する能力が得られるものである。

【0002】

【技術的背景】 オブジェクト指向型システムにおいて、

このオブジェクトとは、データおよびオペレーションを含むコンポーネントであり、これらのオペレーションは、データを操作するために呼出されるようになっている。これらオペレーションは、コールをオブジェクトに送ることによって、このオブジェクトに呼び出される。各オブジェクトはそれぞれタイプを有する。このオブジェクトのタイプが、そのタイプのオブジェクトで実行できるオペレーションを規定する。オブジェクトオペレーションはオブジェクト自身から独立して実行される。さらに、1つのオブジェクトタイプによって、他のオブジェクトタイプ用に規定されると共に実行されるオブジェクトのオペレーションを禁止することが可能となる。これらオブジェクト指向型設計およびプログラム技術に関する更に詳細な説明は、Prentice-Hall社より1988年に発行されたBertrand Mayer (パートランドメイヤー) 著による「Object-oriented Software Construction (オブジェクト指向型ソフトウェア設計)」に記載されている。

【0003】 代表的なクライアント・サーバ演算システムにおいては、一連のコンピュータが設置されており、これらコンピュータ相互は、コンピュータ間を接続しているネットワークを介して通信することができる。これらコンピュータの幾つかは、サービスの提供者として作用するか、または他のコンピュータに対する機能提供者として作用する。このようなサービスまたは機能の提供者は「サーバ」として知られており、これらサービスまたは機能の顧客は、「クライアント」と呼ばれている。また、クライアント・サーバモデルは、同一コンピュータ上で動作している個々のプログラムが、ある保護されたメカニズムを介して互いに通信すると共に、機能の提供者および顧客として動作すると概括することができる。

【0004】 クライアント・サーバモデルによるオブジェクト指向型分散システムでは、オブジェクト指向型インターフェイスをこれらクライアントに提供するサーバが存在する。それらのサーバは、データおよびこれと組合わされたソフトウェアから成るオブジェクトをサポートする。クライアントは、それらのオブジェクトにアクセスすると共に、そのオブジェクトに関するコールを実行してもよい。これらのコールはクライアントからサーバへ転送され、サーバでオブジェクトに組合わされたソフトウェアを介して実行される。これらコールの結果が、次に、クライアントに送り戻される。

【0005】 クライアントプログラムは、オブジェクトハンドル手段によって特定のオブジェクトに参照を付することができる。この「オブジェクトハンドル」とは、オブジェクトを位置決めする方法(ターゲットオブジェクト)およびこのオブジェクトのをコールを実行する方法を与える識別子である。一般に、クライアントプログラムは、動作の開始時に他のプログラムまたは動作中のシ

(3)

特開平 6-19791

システムの核によってオブジェクトハンドルが与えられる。所定のクライアントプログラムが特定のオブジェクトを参照するオブジェクトハンドルを有する場合に、このクライアントプログラムはこのオブジェクトハンドルのコピーを他のクライアントプログラムへ渡すことができる。その第2のクライアントプログラムはオブジェクトにアクセスするためにこのオブジェクトハンドルを用いることができる。

【0006】これらオブジェクトハンドルは種々の方法で実現できる。例えば、1つの既知のシステムは2つのコンポーネントから成るオブジェクトハンドルを利用する。コンポーネントAはサーバのネットワークアドレスであり、コンポーネントBはこのサーバに関するターゲットオブジェクトの識別子である。オブジェクト呼出しが行われると、クライアントはこのオブジェクト呼出しをコンポーネントAによって特定されたネットワークアドレスへ送ると共に、コンポーネントBによって特定された識別子を送る。次に、サーバはコンポーネントBを用いて呼出されている特定のオブジェクトを識別する。他のシステム、即ちエメラルド(Emerald)システムでは、同様に2つのコンポーネントから成るオブジェクトハンドルを利用する。「オブジェクト識別子」と呼ばれる第1のコンポーネントはそのオブジェクトに対する独特な識別子として機能する整数値である。「フォワードアドレス」と呼ばれる第2のコンポーネントは、オブジェクトが最近まで存在しているネットワークアドレスについてのヒントが包含されている。エメラルドクライアントがオブジェクトと通信を所望する場合には、このオブジェクトの現在のネットワークアドレスを決定する必要がある。1988年発行でACM Transactions on Computer Systemsの「Fire-Grained Mobility in the Emerald System」の第109-133頁、Eric Jul, Henry Levy, Norman HutchinsonおよびAndrew Black 著を参照のこと。他のシステムでは、これらオブジェクトハンドルに対して他の種々のフォーマットを利用している。これらオブジェクトハンドルは、本明細書では、「シンプルオブジェクトハンドル」として特徴付けられると共に、これら種々のフォーマットは、以下に説明してあるように、本発明とは無関係である。

【0007】これら「シンプルオブジェクトハンドル」は、分散型システムによってソフトウェア「キャッシング」の完璧な利益が得られるようにするためには、適当なものではない。システムの性能を改善するために「ソフトウェアキャッシュ」が多くのオペレーティングシステムに利用されている。例えば、ファイルキャッシングは、ディスクブロックの内最も最近にアクセスしたもののいくつかをメインメモリ中に保存することによって実現される。ファイルキャッシングは同じデータをディスクからフェッチするときより5~10倍も速く提供することができる。このソフトウェアキャッシングは、分散型

システムにおいても同様の効率が得られる。このキャッシング技術を利用して、クライアントはこのサーバと直接通信せず、むしろキャッシュマネージャプログラムと通信する。このキャッシュマネージャプログラムは、一般に、それ自身用のマシンで動作するものである。このキャッシュマネージャは、サーバの状態の一部を反映させた内部状態を維持するように実行するので、キャッシュマネージャが、このサーバと通信する必要性なく、クライアントのリクエストに応答することができる。この技術によって、実行されるべきクライアントのオペレーションのための時間を減少させることができ、サーバによって実行すべきオペレーションの数も減少できる。

【0008】

【発明が解決しようとする課題】このようなソフトウェアキャッシングをオブジェクト指向型分散型システムで実現でき、ここではオブジェクトを利用してキャッシングが実行される。しかしながら、「シンプルオブジェクトハンドル」の限定条件のために、クライアントマシンレベルでのキャッシングの利益が、或る状況の下ではなくなることがある。例えば、1つのマシンで動作しているクライアントプログラムによって、「シンプルオブジェクトハンドル」を別のマシンで動作している他のクライアントプログラムに送った場合は、いつでも問題が生じる。この問題点については後述する詳細な説明中に開示されている。一般に、第1のプログラムを利用していたオブジェクトハンドルが第1プログラムのマシン上のキャッシュマネージャに対するポインタであり、ターゲットオブジェクト自身に対するポインタでは無いために、この問題が生じる。すなわち、この第1プログラムがそのオブジェクトハンドルを他のマシンのプログラムに渡した場合に、受けとったプログラムは、このオブジェクトハンドルが何処を指示しているか知ることができない。このような状況の下では、以下の理由により非効率が発生してしまう。その理由は、ターゲットオブジェクトのオペレーションの受信中のプログラムの呼出しによって、受信中のプログラムのマシンのキャッシュマネージャを呼び出すか、またはむしろ、オブジェクトハンドルを受信中のプログラムへ送信したプログラムのマシンにおけるキャッシュマネージャの呼び出しが行われる。この不都合は状況によってローカルマシンキャッシングによる利益が完全に無効になる。本発明はこのような問題点を、受信中のプログラムにターゲットオブジェクトに対する追加的な参照を与える送信中のプログラムを持つことによって解決するものである。

【0009】従って、本発明の目的は、オブジェクト指向型の分散型コンピュータシステムにおいて、ソフトウェアキャッシングサービスを可能な限り効果的に利用できる手段を提供することである。

(4)

特開平6-19791

【0010】

【課題を解決するための手段】従って、本発明によれば、増大されたオブジェクトハンドル（実施例中では、「拡張されたキャッシュابلハンドル」と呼ぶ）を有する方法および装置を用意している。この増大されたオブジェクトハンドルによってクライアントに以下の能力が得られる。即ち、クライアントはそれ自身のマシンで動作させているキャッシュマネージャを自動的に選択できる（必要に応じて選択する）。また、このマシンが、異なった種類のオブジェクトをキャッシュ処理する特定化する数種類のキャッシュマネージャを有するならば、この拡張されたキャッシュابلハンドルは、ターゲットオブジェクトの実際の特性に基づいて、キャッシュマネージャの自動選択が実行できる。ローカルキャッシュマネージャが利用できない場合には、受信中のプログラムはターゲットオブジェクトに直接アクセスできる能力を有し、これは送信中のプログラム用のマシンに対して参照作業を行わないでも実現する。

【0011】また、本発明によれば、クライアントプログラムに、キャッシュマネージャおよびターゲットオブジェクトに対する参照が包含されている増大したオブジェクトハンドルを受信するような方法が与えられている。また、受信中のクライアントプログラムには、この増大されたオブジェクトハンドルを変更できる能力が与えられている。すなわち、これを受信したキャッシュマネージャ参照を、受信中のプログラムのマシン上のローカルマネージャに対する参照で差替えるか、またはターゲットオブジェクト自身に対する参照で差替えるかによって変更できる。このことによって、受信中のクライアントプログラムに、変更した増大型のオブジェクトハンドルを利用できる能力が与えられ、これによりターゲットオブジェクトのオペレーションを効率的に呼び出すことができる。

【0012】本発明の一実施例においては、前述した増大型オブジェクトハンドルは拡張されたキャッシュابلハンドルと呼ばれており、このハンドルにはターゲットオブジェクトに対する参照が包含されており、このターゲットハンドルはバックハンドルと呼ばれ、更に、追加の参照が包含されている。これら追加の参照はフロントハンドルおよびキャッシュサービスネームと呼ばれる。この拡張されたキャッシュابلハンドルを受信するクライアントプログラムは、ローカルマシンネーミング（名称化）サービスを利用して、この拡張されたキャッシュابلハンドルのキャッシュサービスネーム部分を、このマシン上のキャッシュマネージャ（即ち、「ローカル」キャッシュマネージャ）上にマッピングする。このローカルキャッシュマネージャは、拡張されたキャッシュابلハンドルのバックハンドル部分によって表示されたターゲットオブジェクトの種類をキャッシングするために有効なものである。次にこのクライアントプログラ

ムはこのバックハンドルを識別したキャッシュマネージャに与えると共に、キャッシュマネージャがそのためのキャッシングサービスを与えるように要求する。このキャッシュマネージャがそのようなサービスを提供できる場合に、キャッシュマネージャ中のローカルオブジェクトを参照するハンドルを戻すようになり、更に、クライアントプログラムは、これを新規なフロントハンドルとして利用する。次に、このクライアントプログラムによって利用するための新規な拡張型キャッシュابلハンドルを構築する。これはバックハンドルとキャッシュサービスネームとを組み合わせることによって実現し、これは元来、上述の新規なフロントハンドルを受信する。キャッシュマネージャが識別されたり、または識別されたキャッシュマネージャがこのキャッシングサービスを実行することを拒否した場合、このクライアントプログラムは、この新規な拡張型キャッシュابلハンドルを、単にバックハンドルをフロントハンドルにコピーするだけで構築する。

【0013】また、他の実施例によれば、この拡張されたオブジェクトハンドルを「シンプルキャッシュابلハンドル」と呼んでいる。このシンプルキャッシュابلハンドルにおいて、ターゲットオブジェクトハンドルをバックハンドルと称し、更に、フロントハンドルと称する僅か1つの追加の参照が存在する。この実施例では、この受信信号のクライアントプログラムは、それ自身のマシン上で適当なキャッシュابلマネージャを選択する固有の能力を持つ必要がある。これに関して、この選択作業は、本例においては自動的に実行される。しかしながら、このローカルキャッシュマネージャを選択した実施例においては、これらフロントおよびバックハンドルを利用した後続の手順が、他の実施例で同一なものとなる。

【0014】

【実施例】

表記法および用語集

以下の詳細な説明を、コンピュータまたはコンピュータのネットワーク上で実行されるプログラム手順によって表現することができる。これらの手順の記載および表現方法は、当事者によって利用される手段であり、この手段によって作業の本質を他の当業者にも最も効果的に伝えることができる。一般、本明細書において、手順としては、所望の結果を導出するためのステップの自己完結型のシーケンスが考えられる。これらステップは、物理量の物理的な操作が必要となるステップである。通常、必ずしも必要ではないが、これら物理的な量は、電気的または磁気的信号の形態をとり、これら信号は記憶したり、伝送したり、結合したり、比較したり、または操作したりできるものである。通常の利用法の理由により、原理的には、これら信号をビットや、値や、エレメントや、記号や、文字や、期間や、数量等として参照するこ

(5)

特開平6-19791

とが望ましいものである。しかしながら、これらすべて用語および同様な用語を適当な物理的な量と組合わせて用いるべきで、および単にこれら量に与えられた便利なラベルとなる。

【0015】更に、実行された操作を場合によっては、人間のオペレータによって行われる精神的なオペレーションである例えば加算や比較のような用語で呼ばれる。人間のオペレータは必要ではなく、多くの場合において好ましくない。本明細書で発明の一部として記載したオペレーションは全てマシンオペレーションである。本発明のオペレーションを実行するための有用なマシンには、汎用のデジタルコンピュータや類似のデバイスが包含されている。また、本発明は、これらオペレーションを実行する装置に関するものである。本発明の装置は、所望の目的のために特に構成されており、これには、汎用のコンピュータが具備されている。このコンピュータは、これに記憶されたコンピュータプログラムによって選択的に作動すると共に変更されるものである。本例で表された手順は、特定のコンピュータや他の装置に固有のものではない。本例で記載された技術に基づいて書かれたプログラムで、種々の汎用マシンを利用できる。または、所望の方法のステップを実行するための更に特別設計された装置を構築することによって、更に便利なものとなる。これらマシンの種々のものに必要な構成は、以下の記載から明らかとなる。以下の記載では、ローカルマシンキャッシングの種々の利点を改善することができる方法によって、オブジェクトと通信できる能力と、オブジェクトを参照して利用する能力とを有するクライアントプログラム間の装置および方法が説明されている。以下の記載では、説明のために特殊なデータおよび構造が開示されており、これによって本発明の全体の理解を助けている。しかしながら、本発明を特別な詳しい情報なしで実現できることは当業者にとって明白である。また他の実施例では、周知のシステムがブロックダイアグラムや概略的な形態で示されており、これによって不必要に本発明を不明確なものにしないようにする。

【0016】先ず、図1のブロックダイアグラムに、分散型コンピュータシステムの代表的なクライアント/サーバが開示されている。ここではクライアントとサーバとはそれぞれ異なったマシンに存在しており、これら両者は、オブジェクトを利用して実行されるキャッシングを採用している。サーバプログラム12は「オブジェクト-02」14を実行するもので、このオブジェクト-02は「ハンドルー-H2」16によって識別される。また、キャッシュマネージャ22は「オブジェクト-01」24を実行し、これは「ハンドルー-H1」26によって識別される。このキャッシュマネージャ22の独特の義務は、オブジェクト-02 14のキャッシングを支持することである。キャッシングの利益を確保しながら、このオブジェ

クト-02にアクセスを望むクライアントは、キャッシュマネージャ22を指示するハンドルー-H1 26を利用する。クライアントによってこのハンドルー-H1 26のオペレーションを発行する場合に、これがキャッシュマネージャ22中のオブジェクト-01 24に伝送される。次に、このキャッシュマネージャ22は、それ自身のオペレーションを実行するか、またはそのハンドルー-H2 16を利用してリクエストを、サーバプログラム12中に存在しているオブジェクト-02 14へ転送する。クライアントプログラムA28は、このハンドルー-H1 26のコピーをクライアントプログラムB30に渡すことができ、またこれらクライアントプログラム28と30とは、オブジェクトハンドルー-H1 26を利用して、オブジェクトにアクセスする。しかしながら、それぞれ異なったマシンに存在するクライアントプログラムが、同一のハンドルー-H1 26を用いて、同一のオブジェクトにアクセスしようとする場合に複雑になる。

【0017】図2を参照しながら説明する。マシンM120のクライアントプログラム28、30が、そのハンドルー-H1 26をマシンM334のクライアントプログラムC32に渡すと、マシンM334のクライアントプログラムC32はマシンM120のキャッシュマネージャ22を利用する。この状況では、キャッシングの種々の利点を損なうようになると共に、クライアントプログラムC32がサーバプログラム12を直接アクセス出来る場合よりも、実際上能率が低下してしまう。マシンM334のクライアントプログラムは、常時、マシンM334上を動作するキャッシュマネージャを通過することが好ましいものである。

【0018】本発明の以下の記載にはこの問題点に対する2つの解決法が開示されている。第1の解決法の一つの実施例は、オブジェクトハンドルを受信するクライアントプログラムが、自分自身のマシンの何れのキャッシュマネージャを各オブジェクト用に利用するかを前もって知った場合に応用するものである。本発明の他の好適な実施例では、適当なキャッシュマネージャの自動的な選択が実行できるようにこの第1解決法を拡大する。これら2つの解決法において、利用されたシンプルオブジェクトハンドルの正確なフォーマットは、本発明に対しては重要なことではない。両方の解決メカニズムは、どのような種類のシンプルオブジェクトハンドルかのシステムで常に利用されているかに基づくものである。例えば、エメラルドシステムにおいては、シンプルオブジェクトハンドルがエメラルドの整数識別子および前進アドレスを含み、一方、上述した他の例では、このシンプルオブジェクトハンドルがネットワークアドレスや関連したオブジェクト識別子を含む。

【0019】この第1の解決法には、「シンプルキャッシュハンドル」と称される新しい種類のオブジェクトハンドルが設定されており、このシンプルキャッシャ

(6)

特開平6-19791

ブルハンドルは、シンプルオブジェクトハンドルより多くの情報を含むように設計されている。この結果、適切なキャッシングが利用できる。シンプルキャッシュャブルハンドルの表示は2つの部分より成り、これら2つの部分は、それら自身がシンプルオブジェクトハンドルである。このシンプルオブジェクトハンドルの第1部分が「フロントハンドル」と呼ばれている。通常、このフロントハンドルによって、ローカルマシン上のキャッシュマネージャ（ローカルキャッシュマネージャと称す）を指示する。また、このシンプルキャッシュャブルハンドルの第2部分は、「バックハンドル」と呼ばれており、これは下に存在するサーバオブジェクト（ターゲットオブジェクト）を常時指示する。また、図3では、典型的なクライアント・サーバ構造が図示されており、ここでは、クライアントプログラムAとB28、30がマシンM130に存在し、クライアントプログラムC32がマシンM334に存在し、さらに、サーバプログラム12がマシンM210に存在している。この図示の例において、クライアントプログラムAとB28、30が表されており、各々には、シンプルキャッシュャブルハンドル58、59が設けられている。開示したシンプルキャッシュャブルハンドル58、59には、フロントハンドル60、61とバックハンドル52、53とが設けられている。これらフロントハンドル60、61はキャッシュマネージャC154中のオブジェクトO156を指示し、このキャッシュマネージャはバックハンドル52を利用してマシンM210上のサーバプログラム12中に位置するオブジェクトO250を指示する。また、クライアントプログラムAとB28、30は、これらのシンプルキャッシュャブルハンドルのバックハンドル52、53部分を利用して、必要に応じてオブジェクトO250を直接指示する。マシンM334上のクライアントプログラムC32はシンプルキャッシュャブルハンドル66を有する。しかしながら、シンプルキャッシュャブルハンドル66のフロントハンドル68は、マシンM120のクライアントプログラムA、B28、30に表されたシンプルキャッシュャブルハンドル58、59のフロントハンドル60、61とは異なるものである。このシンプルキャッシュャブルハンドル66のフロントハンドル68は、マシンM334上に位置したキャッシュマネージャC362中のオブジェクトO364を指示する。このようなシンプルキャッシュャブルハンドルを利用した例において、通常、シンプルキャッシュャブルハンドル上のオペレーションを、フロントハンドルのオペレーション内に向けている。1つのプログラム（受信用プログラムと称す）が、他のプログラムからシンプルキャッシュャブルハンドルを受信する状況の下では、以下の動作が行われる。

【0020】1. フロントハンドルをチェックして、このハンドルが、前述した受信用プログラムが設置してあるマシンと同一のマシン上のキャッシュマネージャを指しているかどうかを決定する。もし、指していれば、

受信したシンプルキャッシュャブルハンドルのフロントハンドルを利用すると共に、これ以上チェックを行わず、さらに残りの動作がスキップされる。

【0021】2. フロントハンドルが、受信用のプログラムが設置されたマシンと同一のマシンのキャッシュマネージャを指示しない場合には、このマシンのキャッシュマネージャを選択する（即ち、ローカルキャッシュマネージャである）。この選択動作は、受信中のオブジェクトハンドルとは独立した静止情報に基づいて実行しなければならない。例えば、ローカルキャッシュマネージャをオブジェクトの予期されたタイプのものに基づいて選択したことが考えられる。

【0022】3. 選択されたローカルキャッシュマネージャがシンプルキャッシュャブルハンドルのバックハンドル部分に与えられると共に、受信用プログラムに対するキャッシングを実行するように要求する。

【0023】4. この選択されたローカルキャッシュマネージャによって、キャッシュマネージャ中のローカルオブジェクトを参照するハンドルが戻されると共に、このハンドルはこのマシン上のシンプルキャッシュャブルハンドル中のフロントハンドルとなる。このローカルキャッシュマネージャは、下に存在するオブジェクトを参照する以前に創作されたオブジェクトを再び利用することができ、または、受信用プログラム向きのこのキャッシュオブジェクトを創作することもできる。

【0024】5. 次に、この受信用プログラムによって新たなシンプルキャッシュャブルハンドルが構築され、このハンドルはローカルキャッシュマネージャプログラムからのフロントハンドルと、受信用プログラムへ送信したバックハンドルとから構成される。

【0025】また、上述のステップ2～5まで、すべて失敗すればこの受信用プログラムは、バックハンドルを単にフロントハンドルとして使用する。従って、所定のマシンが適当なキャッシュマネージャを有しない場合、またはこのキャッシュマネージャが協働しない場合には、受信用プログラムは、以下の条件であったとしても下に存在するサービスと未だ通信することが可能となる。即ち、ローカルマシン上にキャッシングの利益が存在しないのみならず、図2で示したように送信用マシン上にキャッシュマネージャを不必要に使うことによって生じる欠点が存在しなくても可能となる。この受信用プログラムによって、シンプルキャッシュャブルハンドルを他のアプリケーションプログラムへ再び送信した場合に、このアプリケーションは、それ自身のマシンの適当なキャッシュマネージャとコンタクトを取る機会を再び持つようになる。

【0026】要約すると、シンプルキャッシュャブルハンドルは、常に、同一の下側に存在するオブジェクトを参照するように変更されるオブジェクトのカテゴリであるが、これらのハンドルを他のアプリケーションプログラ

(7)

特開平6-19791

ムへ送信した時は、そのハンドルを異なったキャッシュマネージャを利用するように再び仕向けられる。フロントハンドルは変更可能であるが、バックハンドルは一定のままである。

【0027】上述した本発明の択一的な実行には、これには、どのキャッシュマネージャがそれを利用することを望んでいるかを決定するシンプルキャッシュابلハンドルを受信するプログラムを必要とする欠点がある。しかしながら、マシンには、異なった種類のオブジェクトを、オブジェクトの位置または、利用しているキャッシングアルゴリズムに基づいて特定化している数種類のキャッシュマネージャが設けてある。従って、本発明の一実施例では、「拡張されたキャッシュابلハンドル」と称する新しい種類のオブジェクトハンドルを採用している。この新しいオブジェクトハンドルによって、ローカルキャッシュマネージャの自動的な選択が、ターゲットオブジェクトの実際の特性に基づいて実行できるように設計されている。

【0028】この拡張されたキャッシュابلハンドルの表示は3つの部分より構成されており、これら部分の2つは前述したシンプルオブジェクトハンドルである。拡張されたキャッシュابلハンドルの第1部分は、「フロントハンドル」と呼ばれ、これは通常、ローカルマシン上のキャッシュオブジェクトを指示する。また、このハンドルの第2部分は、「バックハンドル」と呼ばれ、これによって事実上下側に存在するオブジェクト（ターゲットオブジェクト）を指す。また、この拡張されたキャッシュابلハンドルの第3部分は、「キャッシュサービスネーム」または、単に「キャッシュネーム」と呼ばれる。このキャッシュネームによって、キャッシングサービスの種類を識別し、このサービスによってこの特定のオブジェクト用のキャッシングをどのように管理するかを理解する。前述したシンプルキャッシュابلハンドルの場合のように、拡張されたキャッシュابلハンドルは、転送中に変更できるオブジェクトハンドルのカテゴリであるので、その結果、これら拡張されたキャッシュابلハンドルは、常時、同一の下側に存在するオブジェクトを参照するようになるが、これらハンドルを他のアプリケーションプログラムへ転送した時は何時でも、異なったキャッシュマネージャを利用するために再び仕向けることができる。このフロントハンドルを変更できるが、バックハンドルおよびキャッシュサービスネームは一定のままである。

【0029】一実施例において、拡張されたキャッシュابلハンドルが、下側に存在するオブジェクトが創作されたときに、サーバプログラムによって創生される。キャッシュサービスネームおよびバックハンドルがサーバプログラムによって規定され、またこのサーバプログラムによって、当所からフロントハンドルがバックハンドと等価となるようにセットされる。また、一実施例にお

いて、各マシンによってネーミングサービスが得られる。このネーミングサービスによってキャッシュサービスネームからキャッシュマネージャプログラムへプログラムがマッピングされる。このキャッシュマネージャプログラムは、このマシン上で動作されると共に、キャッシングの所定のスタイルを支持することが可能となる。キャッシュマネージャプログラムが開始されると、これらプログラム自身がこのネーミングサービスと一緒に、キャッシュサービスネームを特定のキャッシュマネージャプログラムに関連付ける用法で登録される。この結果、マシンネーミングサービスを利用することによって、キャッシュマネージャプログラムをこのマシンで動作するクライアントプログラムにより見つけることが可能となる。これら拡張されたキャッシュابلハンドルの利用が図4に表示されている。

【0030】ここで、図4を参照しながら、クライアント・サーバの代表的な構成を説明する。ここでは、クライアントプログラムAとB28、30がマシンM120に存在し、クライアントプログラムC32がマシンM334上に存在し、更にマシンプログラム12が、マシンM210に存在する。この図4では、各々が拡張されたキャッシュابلハンドル70、71を有するクライアントプログラムA、B28、30が図示されている。これら拡張されたキャッシュابلハンドル70、71はフロントハンドル60、61と、バックハンドル52、53と、キャッシュサービスネーム（キャッシュネーム）76、77とを有している。これらキャッシュネーム76、77はクライアントプログラムA、B28、30によって利用されて、ネーミングサービス80からキャッシュマネージャC154へのポイントを得ていた。次に、クライアントプログラムAとB28、30はキャッシュマネージャC154と通信しており、これはバックハンドル52、53を送信し、フロントハンドル60、61を応答としてそれぞれ受信する。これらフロントハンドル60、61によってキャッシュマネージャC154中のオブジェクトO156を指示する。このマネージャはバックハンドル51を利用して、マシンM210のサービスプログラム12に存在するオブジェクトO250を得ている。また、クライアントプログラムA、B28、30は、それらの拡張されたキャッシュابلハンドルのバックハンドル52、53部分を用いて、必要に応じてオブジェクトO250を直接指示することができる。マシンM334のクライアントプログラムC32は、また拡張されたキャッシュابلハンドル74を有している。しかしながら、このハンドル74のフロントハンドル68は、マシンM120のクライアントプログラムAとB28、30に示した拡張されたキャッシュابلハンドル70、71のフロントハンドル60、61とは異なる。拡張されたキャッシュابلハンドル74のフロントハンドル68は、マシンM334に存在するキャッシュマネージャC362におけるオブジェクトO364を指示する。このフロントハンドル68は、クライアントプログラムC

(8)

特開平6-19791

32によって以下の方法によって得られた。即ち、最初、マシンM334上のネーミングサービス82を利用して、キャッシュサービスネーム79をローカルキャッシュマネージャC362へのポイントに変える。次に、プログラムC32とキャッシュマネージャC362とを、バックハンドル55を送信すると共に、フロントハンドル68を応答として受信することにより通信していた。

【0031】拡張されたキャッシュابلハンドルを利用した例において、通常、この拡張されたキャッシュابلハンドルのオペレーションは、フロントハンドルのオペレーションに向けられる。図5において、プログラムが他のプログラム100からの拡張されたキャッシュابلハンドルを受信（即ち、受信用プログラム）する状況の下で、以下の動作が実行される。

【0032】1. フロントハンドルをチェックして、このハンドルがステップ102で受信用プログラムが載っているマシンと同一マシン上のキャッシュマネージャを指示するかどうかを決定する。ステップ104で指示(YES)であれば、受信した拡張型キャッシュابلハンドルのフロントハンドルを用いると共に、これ以上のテストを行わず、残余の動作をスキップする。

【0033】2. 若し、このフロントハンドルが同一マシン106のキャッシュマネージャを指示しない(NO)場合には、受信したプログラムはそのマシンに関するネーミングサービスを利用して、拡張されたキャッシュابلハンドルのキャッシュサービスネーム部分をローカルキャッシュマネージャに対する参照部分にマッピングする。このローカルキャッシュマネージャは、この拡張型キャッシュابلハンドル108によって表されたオブジェクトの種類を扱うように設置されている。

【0034】3. このような参照がステップ112で利用できる場合に、次に、受信用プログラムによって、表示されたローカルマネージャがこの拡張型キャッシュابلハンドルのバックハンドル部分と一緒に表されると共に、このキャッシュマネージャがステップ114で受信用プログラム用のキャッシングを行うように要請する。

【0035】4. このローカルキャッシュマネージャがそのようなサービスを提供することに同意した場合116に、このキャッシュマネージャはハンドルを戻す。このハンドルによって、受信用プログラムに対するキャッシュマネージャ中のローカルオブジェクトを参照する。このプログラムは、それ自身の拡張されたキャッシュابلハンドル中にこの参照事項をフロントハンドルとして利用するようになる。このキャッシュマネージャによって、下側に存在するオブジェクトを参照する以前に創作されたオブジェクトを再度利用することができ、または、このキャッシュオブジェクトを受信用プログラム用に創作することもできる。

【0036】5. 次に、受信用プログラムによって、キャッシュマネージャから得たフロントハンドルをバック

ハンドルおよびキャッシュサービスネームに追加する。これらバックハンドルおよびキャッシュサービスネームは受信用プログラムへ送信されており、これにより訂正された拡張型キャッシュابلハンドル表示122を構成する。

【0037】6. 若し、この手順のステップ3または4で、NOならば、この受信用プログラムは、このバックハンドルを単に、フロントハンドル120として利用する。次に、所定のマシンが適当なキャッシュマネージャ110を有しない場合、またはこのキャッシュマネージャが協働作業しなければ、受信用プログラムは依然として下側のオブジェクトと通信可能となる。これは、ローカルキャッシングの利益を損なうのみならず、他のマシンと不必要に通信する不利益がなくても行う。受信用プログラムによって、この拡張型キャッシュابلハンドルが他のアプリケーションプログラムへ送信されると、このアプリケーションプログラムは、それ自身のマシン上の適当なキャッシュマネージャとコンタクトを取る機会に恵まれる。

【0038】拡張型キャッシュابلハンドルを利用した実施例およびシンプルキャッシュابلハンドルを利用した他の実施例の両者において、要求があった場合には、何時でもこれらのハンドルのすべてのコンポーネントを送信する。また拡張型キャッシュابلハンドルまたはシンプルキャッシュابلハンドルのいずれか一方を用いる本発明の他の実施例では、フロントハンドルの送信を回避することができる。この場合、受信用プログラムは常時、適当なフロントハンドルを見つける責任を負うものである。これは、上述した手順のいずれかのステップ1をスキップすることにより達成される。

【0039】上述したように、本発明を好適な実施例を利用して説明したが、本発明は上述した実施例のみに限定されず、種々の変更を加え得ることは明白である。従って、上述の記載によって本発明の趣旨が限定されるものでなく、請求の範囲に記載された技術的思想に基づいて決められるものである。

【図面の簡単な説明】

【図1】従来のシンプルオブジェクトハンドルを利用したオブジェクト指向型の分散型システムのソフトウェアキャッシング方法を示す線図。

【図2】従来のシンプルオブジェクトハンドルを用いた場合の問題点を示す線図。

【図3】本発明のシンプルキャッシュابلハンドルを利用した分散型システムを示す線図。

【図4】同じく、拡張されたキャッシュابلハンドルを利用した分散型システムを示す線図。

【図5】本発明の一実施例の動作を示すフローチャート。

【符号の説明】

12サーバプログラム

(9)

特開平6-19791

14、24 オブジェクト

16、26、58、59、60、61 ハンドル

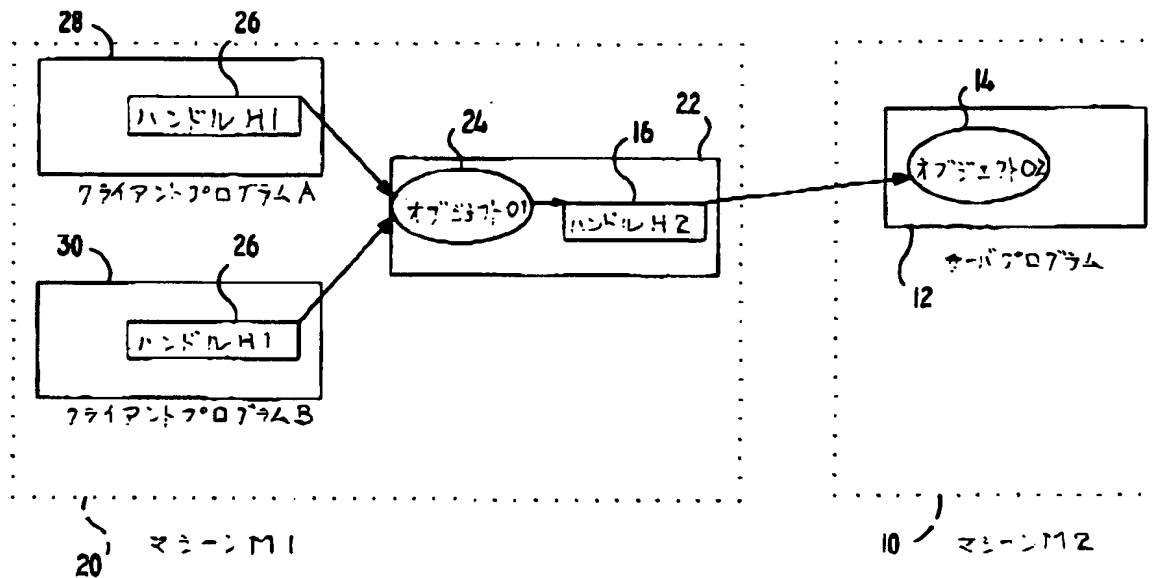
22 キャッシュマネージャ

28、30、C32クライアントプログラム

10、20、34 マシン

62 ローカルキャッシュマネージャ

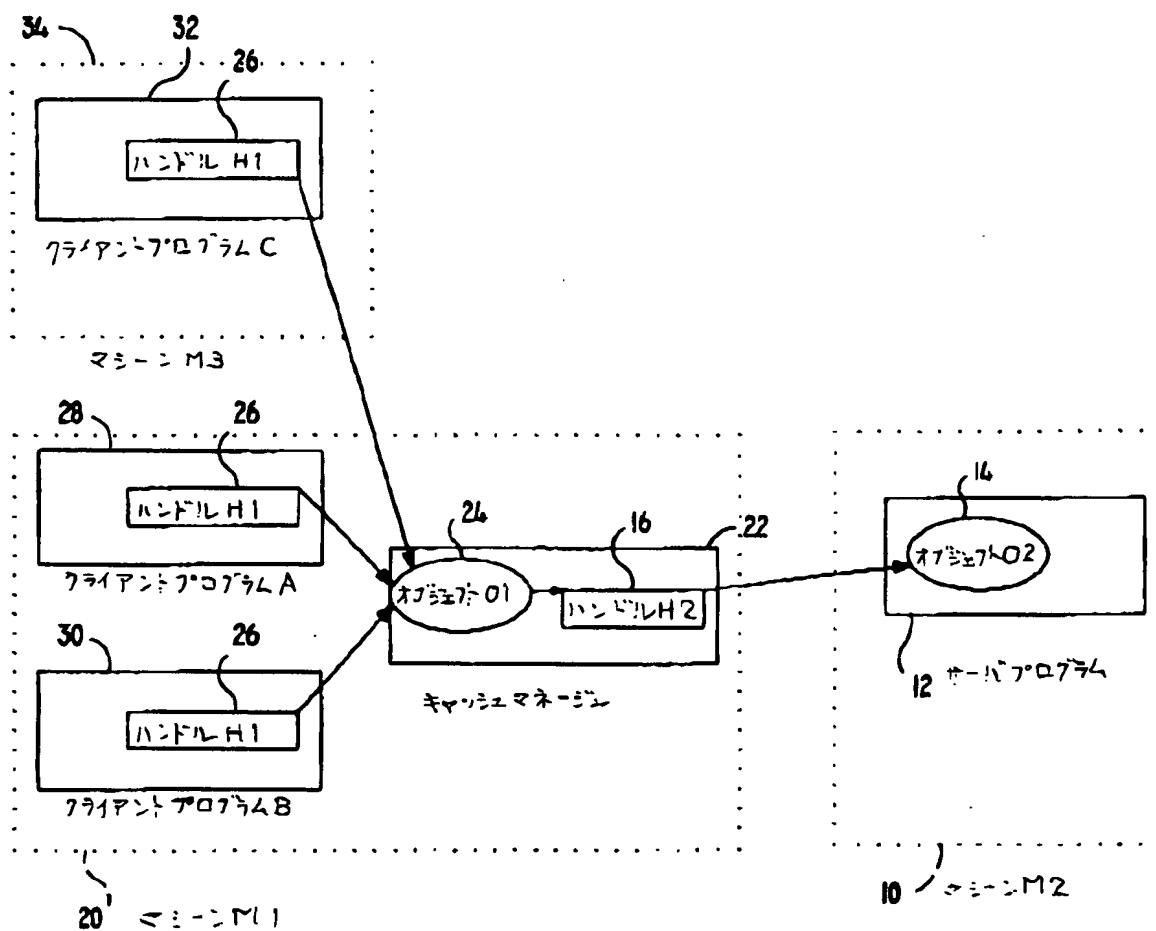
【図1】



(10)

特開平6-19791

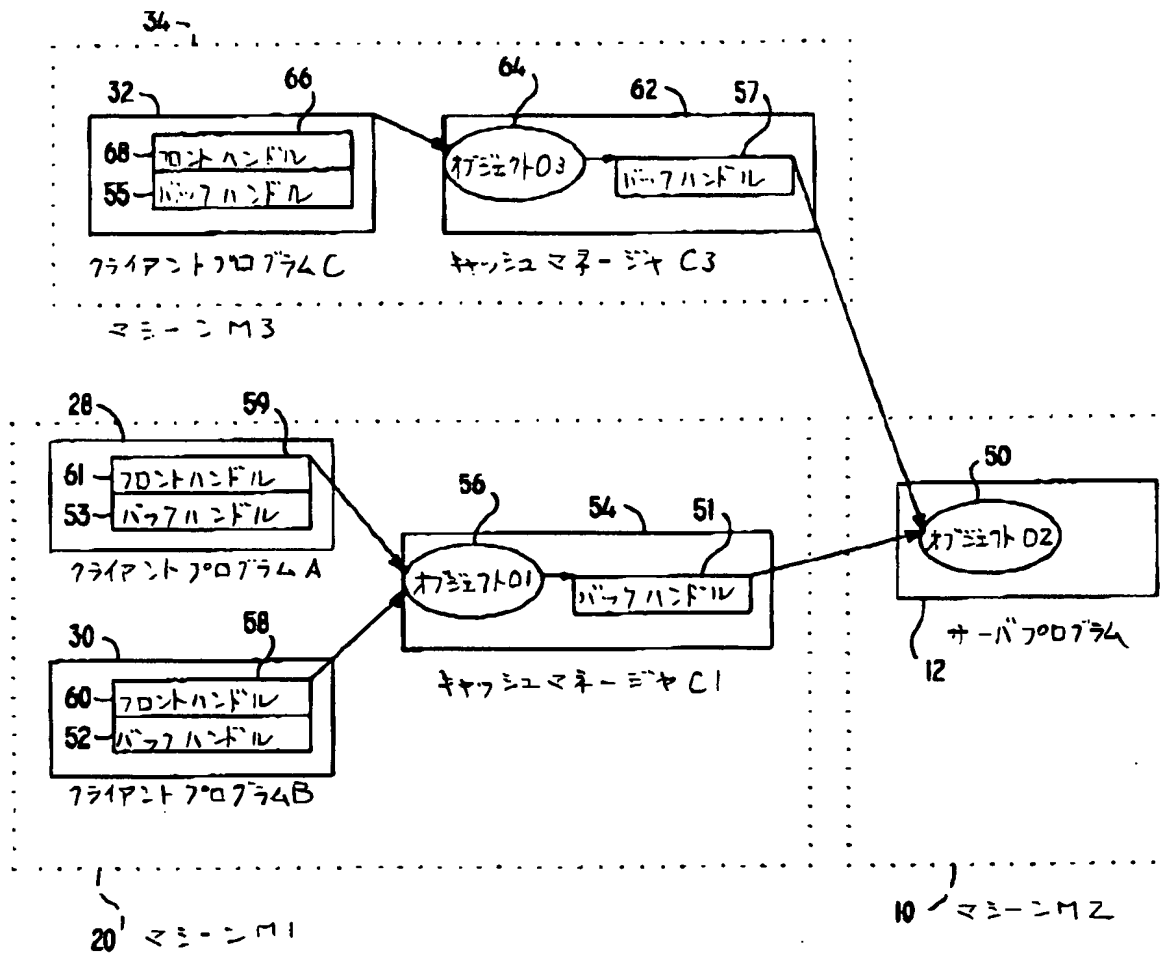
【図2】



(11)

特開平6-19791

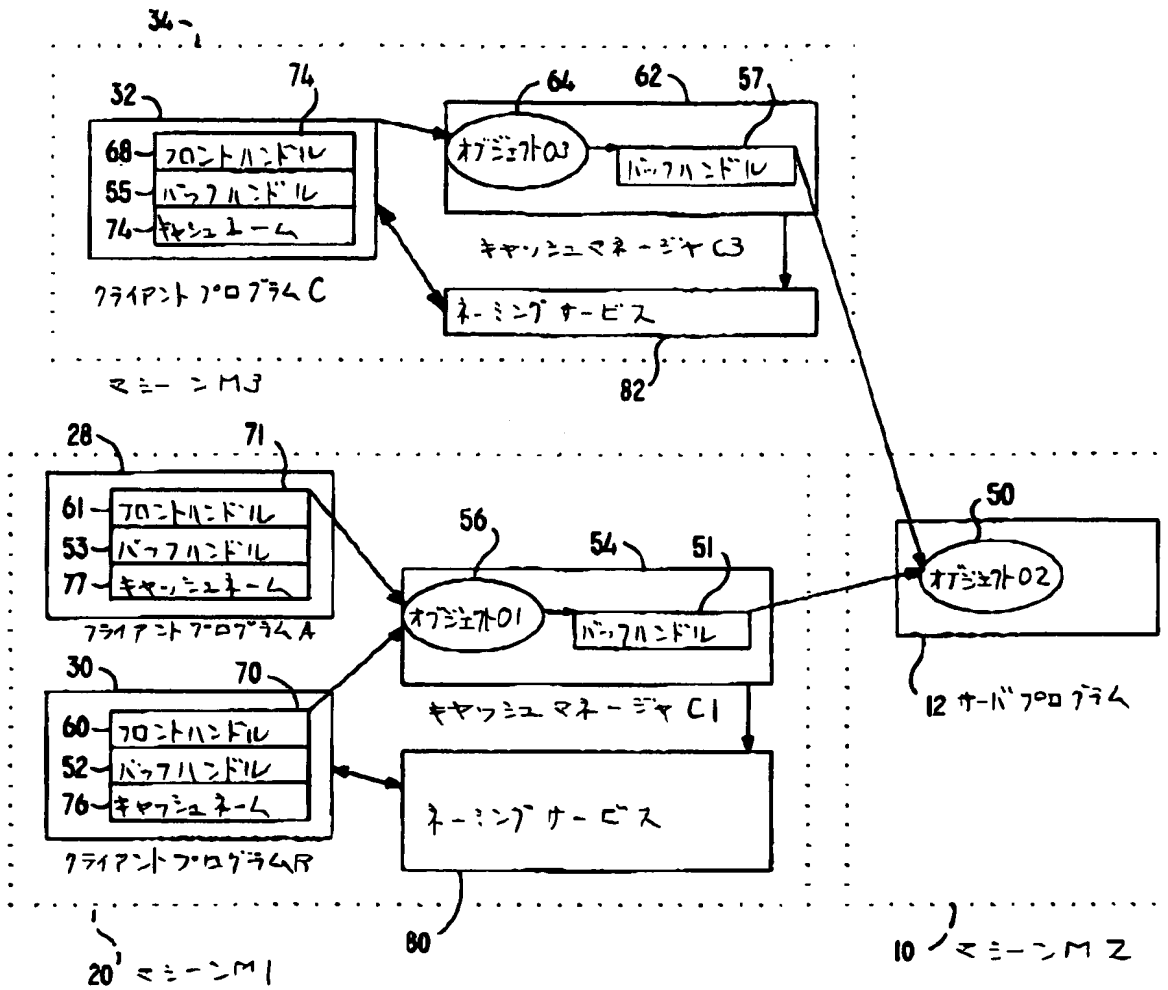
【図3】



(12)

特開平6-19791

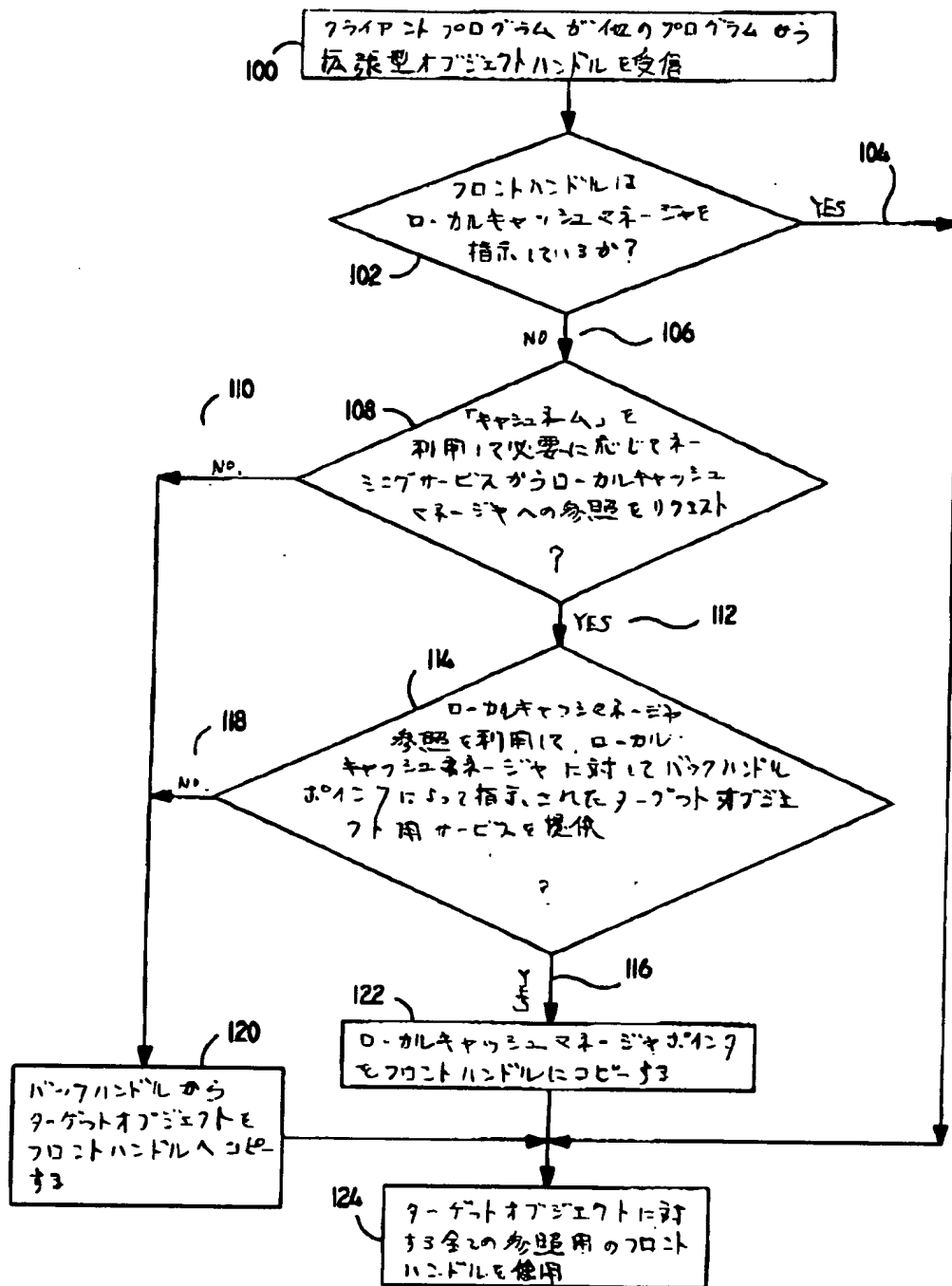
【図4】



(13)

特開平6-19791

【図5】



フロントページの続き

(72)発明者 マイケル・エヌ・ネルソン
 アメリカ合衆国 94070 カリフォルニア
 州・サン カルロス・ヒルクレスト ロー
 ド・158